FakeBlock: Blockchain-based DApp to counter Fake News

CS765: Introduction to Blockchains, Cryptocurrencies and Smart Contracts

Assignment report by,

Prathamesh Chandrakant Navale (23M0746)
Varn Gupta (23M0749)
Abhishek Dilip Jagushte (23M0791)

Department of Computer Science and Engineering
Indian Institute of Technology, Bombay

Apr 14, 2024

Section 1

In today's information-rich environment, distinguishing between accurate news and false information has become increasingly difficult. Our blockchain-powered Decentralized Application (DApp) tackles this issue by utilizing the Proof of Stake algorithm and a strong consensus protocol. This system ensures that nodes disseminating fake news not only lose their trustworthiness but also face a financial penalty. This dual mechanism not only promotes honest participation but also discourages dishonest behavior, making our platform an effective tool in the battle against misinformation.

Below are the factors we are considering while designing the application:

1.1 Sybil Attack

A Sybil attack is a form of malicious activity in a peer-to-peer network where a single user creates multiple fake identities, called Sybil nodes, to gain a disproportionately large influence over the network. This can be used to manipulate the network's behavior, such as spreading false information, disrupting communication, or controlling resources.

We counter the sybil attack using **Proof of Stake**. Every node in the system has a balance initialized at the start of the simulation. Knowing the balances of all the nodes in the system can be done robustly in a blockchain. Now, while voting we consider the node balances as a weight towards the aggregated voting result. This ensures that nodes with more balance have more say in the vote.

By doing this, we ensure that the malicious party cannot create multiple fake identities and participate in the vote to sway the results.

Below is how we are calculating the voting result

$$result = [\sum vote_i * (credibility_i + k * balance_i + e)/\sum (credibility_i + k * balance_i + e)] > 0.5$$

...(i)

where e is decided based on expertise. More about this in section 1.3.

1.2 Voter trustworthiness evaluation

We need to evaluate the voter trustworthiness after each vote based on what they voted for and what is the result of the vote. We decrease the credibility of the voter if they have voted opposite to the result of the voting.

In case the voter is correct, then

$$credibility = credibility + CRDIBILITY\_INC\_MULTIPLIER * (100 - credibility)$$

This also ensures that the credibility value does not exceed 100. For the case when the voter is incorrect, we penalize the voter's credibility
$credibility = credibility - CRDIBILITY\_INC\_MULTIPLIER * (100 - credibility)$

Thus the voters who have voted incorrectly will lose their weight in upcoming votes whenever they are selected as a fact checker.

1.3 Voter Expertise

We have defined different fields of news in which voters can have expertise. Expertise can be decided by tallying how many correct votes the user has previously given on some category of news. Below we explain upon e used in *equation (i).*

$e = news.category == voter.expertise\ ?\ EXPERTISE\_MULTIPLIER : 0$

So we add a weight of 20% of the voter credibility if they have expertise in the topic of the news. This ensures that expert voters have more say in the vote about a particular category of news.

1.4 Rational Voter Incentivization

Voters who vote correctly need to be incentivized so that they have the benefit of being honest. This will encourage organizations to be fact-checkers and in that, honest fact-checkers.

We collect the reward to be given in each vote from the following sources
1. The fact verifying fee from the users who want to verify the news from the system.
2. The penalty imposed on the voters who have voted incorrectly.

The total reward is then equally distributed to voters who had voted correctly.

1.4 Uploading a news item

We assume that anonymous parties upload news items to our system. These news items are then voted on by the consensus committee that is elected randomly.

1.5 Bootstrapping

Every user registering in the system has a base credibility. This value is such that it won't be too advantageous in the vote. The user will have to participate in more polls with correct votes to earn credibility.

Section 2 - Analysis of Simulation

Analysis:

The analysis of the simulation results provides insights into the behavior of DApp under different conditions.

2.1 Effect of Number of Nodes

Varying the number of nodes to check its effect on the network and the trustworthiness of the nodes.
In this, we kept p = q = 0..2 :

Number of Nodes	**AVG_bal_H_0.9**	**AVG_bal_H_0.7**	**AVG_bal_M_0**
10	133.829	96.479	0.004
50	121.572	99.791	0.011
100	114.796	100.825	0.524

Number of Nodes	**AVG_cred_H_0.9**	**AVG_cred_H_0.7**	**AVG_cred_M**
10	97.054	60.233	6.790
50	96.316	53.660	0.338
100	86.015	50.839	0.000

The number of nodes affects the distribution of balances among nodes, both malicious and honest. With an increase in the Number of Nodes, the balance and credibility of the honest nodes decreases, and the balance and credibility of the malicious nodes slightly increase which indicates that the malicious nodes can exploit a larger network more effectively.

2.2 Effect of q (fraction of malicious nodes)

The effect of q on the average balance and credibility of the nodes in the network.

q (fraction of M)	**AVG_bal_H_0.9**	**AVG_bal_H_0.7**	**AVG_bal_M**	**%age of stake held by malicious nodes**
0.2	113.967	101.048	0.471	3.6%
0.51	124.577	112.239	0.870	13.5%
0.9	87.763	90.974	16.230	57.4%

q (fraction of M)	AVG_cred_H_0.9	AVG_cred_H_0.7	AVG_cred_M
0.2	86.115	49.983	0.000
0.51	81.008	51.373	0.054
0.9	3.388	18.181	100.000

The fraction of malicious nodes affects a lot. Even though our simulation can avoid a 51% attack (initially malicious nodes are given fewer bitcoins than honest nodes; **malicious nodes got 15 bitcoins while honest got 100)** still when a fraction of malicious nodes is 0.9, malicious can take over as it possible most of the time only malicious are voting for the news. But since there's a deposit for the voting, malicious nodes won't be able to divide infinitely and have to buy more bitcoins.

2.3 Effect of p (fraction of nodes with prob 0.9 in Honest nodes)

Varying the fraction of Honest nodes with prob 0.9 affects the network and the voting system.

p (fraction of H_0.9)	AVG_bal_H_0.9	AVG_bal_H_0.7	AVG_bal_M
0.2	113.967	101.048	0.471
0.4	109.970	99.268	0.804
0.9	105.358	87.183	0.839

p (fraction of H_0.9)	AVG_cred_H_0.9	AVG_cred_H_0.7	AVG_cred_M
0.2	86.115	49.983	0.000
0.4	83.185	52.466	0.000
0.9	81.623	41.841	0.000

Higher values of "p" lead to a more equitable resource distribution among honest nodes but may also result in decreased credibility scores for individual highly trustworthy nodes. Malicious nodes struggle to accumulate significant resources within the decentralized system, suggesting the effectiveness of security measures against malicious behavior.

Conclusion

Using proof of stake we can successfully avoid sybil attack on the decentralized news verification application built on top of blockchain. We have successfully shown that even if the malicious nodes possess 51% of the voting power, they cannot win the vote if they don;t have over 20% wealth (balance) in the system. Hence the Proof of Stake method works and potentially can solve the problem of fake news propagation!